

## Deliverable-03

Lint Report      Workgroup: E7.04      Date: 25/04/2022

### **Miembros:**

Baños González, Alejandro ([alebangon@alum.us.es](mailto:alebangon@alum.us.es))

Flores Rodríguez, Román ([romflorod@alum.us.es](mailto:romflorod@alum.us.es))

Grosso Gómez de Terreros, Javier ([javgrogom@alum.us.es](mailto:javgrogom@alum.us.es))

Gutiérrez Ceballos, Pablo ([pabgutceb@alum.us.es](mailto:pabgutceb@alum.us.es))

Ibáñez Montero, Julia ([julibamon@alum.us.es](mailto:julibamon@alum.us.es))

Roldán Cadena, Jesús ([jesrolcad@alum.us.es](mailto:jesrolcad@alum.us.es))

**Repositorio de Github:** <https://github.com/jesrolcad/Acme-Toolkits>

## Índice

1. Resumen ejecutivo	1
2. Tabla de control de cambios	2
3. Introducción	3
4. Resultados obtenidos	4
5. Cambios realizados	5
6. Conclusiones	6
7. Referencias bibliográficas	7

# 1. Resumen ejecutivo

Este documento detalla los resultados obtenidos tras el análisis del código con la herramienta SonarLint. Para ello, se muestran capturas de los resultados y se especifica la solución que hemos proporcionado para eliminar por completo los malos olores de nuestro código.

Hemos observado que los malos olores se han concentrado en los distintos dashboards. En concreto, hemos cometido fallos al hacer uso de las tablas en los form.jsp de dichos dashboards, ya que hemos utilizado etiquetas que han quedado obsoletas y hemos omitido algunos atributos importantes dentro de las tablas.

Hemos concluido que el trabajo realizado para deshacernos de los malos olores no nos ha supuesto una gran carga de trabajo ya que, aunque en repetidas ocasiones, solo hemos obtenido tres malos olores del análisis del código con SonarLint. Además, Eclipse nos proporciona algunas herramientas (por ejemplo, replace all) que nos han facilitado el hacer los cambios oportunos sobre el código.

Consideramos que, aunque esta tarea no nos ha supuesto mucho tiempo ni esfuerzo, habría sido una buena práctica el haber llevado un seguimiento más periódico de los malos olores del código, ya que de esta manera no habríamos cometido los mismos errores una y otra vez durante el desarrollo.

## 2. Tabla de control de cambios

<b>Número de revisión</b>	<b>Descripción</b>	<b>Fecha</b>
1	Creación del documento y sus apartados	23/04/2022
2	Redacción completa del documento	24/04/2022

### 3. Introducción

En este documento se reflejan los resultados obtenidos tras hacer un análisis del código con la herramienta SonarLint. Se van a mostrar capturas de los malos olores detectados por la herramienta, así como una explicación de los cambios que hemos realizado para solucionarlos. Para ello, hemos organizado este documento en distintas secciones descritas a continuación:

**Sección 4. Resultados obtenidos.** En esta sección se muestran capturas del análisis de nuestro código proporcionado por SonarLint.

**Sección 5. Cambios realizados.** Se explican los cambios realizados tras el análisis del código, así como una captura final para mostrar la solución a los malos olores.

**Sección 6. Conclusiones.** Se reflejan las conclusiones que hemos obtenido del análisis del código con SonarLint y se hace un balance de la metodología que hemos seguido para la detección y eliminación de malos olores.

**Sección 7. Referencias bibliográficas.** Esta sección está vacía ya que no hemos recurrido a ningún tipo de bibliografía para la redacción de este documento.

Finalmente, el documento posee la siguiente estructura: portada, índice, resumen ejecutivo, tabla de control de cambios, introducción, contenidos (divididos en las secciones *Resultados obtenidos* y *Cambios realizados*), conclusiones y referencias bibliográficas.

## 4. Resultados obtenidos

Tras hacer el análisis de nuestro código (finalizado) con la herramienta SonarLint, hemos obtenido un total de 132 malos olores. Dichos malos olores se encuentran todos en el form.jsp de patron-dashboard y de administrator-dashboard. Todos los malos olores obtenidos se repiten varias veces por lo que se pueden resumir en un total de tres malos olores.

En primer lugar, falta el atributo description en las tablas:





















Resource	Date	Description
form.jsp		🐞 Add a description to this table.
form.jsp		🐞 Add a description to this table.
form.jsp		🐞 Add a description to this table.
form.jsp		🐞 Add a description to this table.
form.jsp		🐞 Add a description to this table.
form.jsp		🐞 Add a description to this table.
form.jsp		🐞 Add a description to this table.
form.jsp		🐞 Add a description to this table.

En segundo lugar, falta el atributo id o scope en las tag <th>:

Resource	Date	Description
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.
form.jsp		🐞 Add either an 'id' or a 'scope' attribute to this <th> tag.

Por último, se ha utilizado el atributo “align” de la siguiente manera (cuando está obsoleto):

```
<td align="right">
```

Resource	Date	Description
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.
form.jsp		  Remove this deprecated "align" attribute.

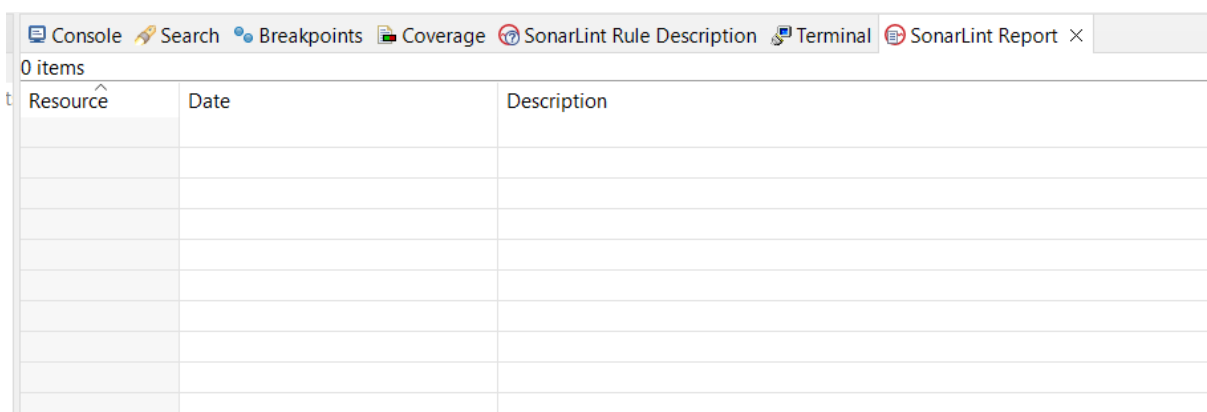
## 5. Cambios realizados

Para paliar el primer mal olor (atributo description ausente), hemos añadido en cada tabla del form.jsp, para el patron-dashboard y el administrator-dashboard, la etiqueta `<caption>`.

Para el segundo mal olor, en el que se nos especificaba que no teníamos puesto el atributo id para cada etiqueta `<th>`, lo que hemos hecho ha sido añadir el id a cada una de esas etiquetas de la siguiente manera: `<th id="">`.

Para el último mal olor, hemos sustituido la etiqueta `<td align="right">` por `<td style="text-align:right;">`, ya que esta última no está obsoleta y es más recomendable.

Como podemos ver en la siguiente captura, todos los malos olores han desaparecido gracias a estos cambios:



0 items		
Resource	Date	Description

## 6. Conclusiones

Tras haber hecho el análisis del código con SonarLint hemos concluido que solo en una parte muy concreta del proyecto hemos tenido malos olores (en los dashboards), por lo que el trabajo de eliminarlos no ha sido una complicación. Además, con la herramienta replace de eclipse, hemos conseguido hacer este trabajo sin que nos suponga una inversión de tiempo muy grande.

Gracias al análisis que nos ha proporcionado SonarLint hemos aprendido varias cosas sobre el uso de algunas etiquetas en jsp:

En primer lugar, hemos aprendido la importancia que tiene poner siempre el atributo id en las etiquetas <th> ya que puede ser muy útil para identificar esa parte del código y diferenciarla de los demás <th> por si en un futuro tenemos que implementar alguna funcionalidad que afecte a ese código (el perteneciente a dicha etiqueta <th>).

También, hemos aprendido que el atributo align tal y como lo estábamos utilizando no es recomendable ya que ha quedado obsoleto, por lo que es mucho mejor utilizar <td style= "text-align:right;">.

En cuanto al atributo description, hemos comprendido la importancia que tiene añadir una la etiqueta <caption> para poder introducir una descripción a las distintas tablas. En nuestro caso, hemos dejado las etiquetas vacías ya que con los títulos de las tablas consideramos que la comprensión del dashboard hace que no sea necesario especificar ninguna descripción. Aun así, hemos entendido que SonarLint nos haya recomendado ponerlas, ya que si en algún momento queremos añadirle una descripción a una tabla podemos hacer uso de esta etiqueta.

Nuestra metodología para la detección de malos olores ha sido ejecutar SonarLint sobre la rama master en el momento en el que todas las funcionalidades ya habían sido implementadas. Consideramos que habría sido mejor llevar un seguimiento periódico de los malos olores para no cometer los mismos fallos una y otra vez durante el desarrollo. En nuestro caso, si hubiésemos detectado el mal uso que le estábamos dando a algunas etiquetas de las tablas del dashboard para el usuario administrador, no habríamos cometido los mismos fallos para el dashboard del usuario patron, por lo que habríamos tenido que cambiar menos líneas de código al final del sprint.



## 7. Referencias bibliográficas